UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/521,586	09/26/2005	Joachim Roos	1533-1004	4031

466   7590   04/08/2008
YOUNG & THOMPSON
209 Madison Street
Suite 500
ALEXANDRIA, VA 22314

EXAMINER
VICARY, KEITH E

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
04/08/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/521,586	ROOS ET AL.
	Examiner	**Art Unit**	
	Keith Vicary	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 February 2008*.
2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) *1-20* is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
      1.☐ Certified copies of the priority documents have been received.
      2.☐ Certified copies of the priority documents have been received in Application No. _____.
      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.
4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5)☐ Notice of Informal Patent Application
6)☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1.   A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 2/19/2008 has been entered.

2.   Claims 1-20 are pending in this application and presented for examination. Claims 1, 8, and 12 are newly amended and claims 15-20 are newly added by amendment filed 2/19/2008.

Claim Objections

3.   Claim 15 is objected to because of the following informalities. Appropriate correction is required.

        a.   Claim 15 recites the limitation "according to claim 1, fair manner" which should presumably be "according to claim 1, wherein the fair manner".

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 1-16 and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cataldo (Net processor startup takes pipelined path to 40 Gbits/s) in view of Dorst (20040098549 A1) in view of Harriman (US 6330645 B1).

6.   Consider claims 1 and 8, Cataldo discloses a programmable pipeline adapted to directionally transfer data packets through the pipeline from a first end of the pipeline to a second end of the pipeline, and adapted to perform sequences of instructions on the data packets (Page 1, second to fifth paragraphs, programmable pipeline architecture...PISC can be programmed to do tasks...each PISC stage acts as a mini-processor). Cataldo also inherently discloses of plural access points located in a spaced apart relation between the first end of the pipeline and the second end of the pipeline (as, read broadly, these access points could simply be ports to each stage or the like).

However, Cataldo does not disclose at least one interface engine adapted to be connected to at least one external device located externally of the processor; the at least one interface engine is connected to each of the plural access points, and wherein the at least one interface engine is adapted to receive a request from any one of the plural access points of the programmable pipeline, the request being received upon arrival of one of the data packets at the respective any one access point, ii) to send a request output to the external device, the request output being based at least partly on the request from the one access point, iii) responsive to the request output, to receive

an external reply from the external device, and iv) to send to the pipeline a response, based on the external reply, to the request. Cataldo also does not disclose that the interface engine includes an arbiter configured to allow forwarding of requests from the plural access points in a fair manner between each of the plural access points.

On the other hand, Dorst does disclose at least one interface engine (figure 3, memory controller 1005, which includes interface circuitry 4010 as shown in figure 4) adapted to be connected to at least one external device located externally of the processor (figure 3 shows multiple memories 1015 external to the processor and connected to the data processing block 3005, which includes the memory controller); the at least one interface engine is connected to each of the plural access points, ([0031], several processors may share a memory controller; each processor is an access point or contains an access point), and wherein the at least one interface engine is adapted to i) receive a request from any one of the access points ([0010], memory controller couples to the processors and to the memories, and provides communication between the processor and memories; it is inherent that this communication includes a request for data in a memory by the processor), ii) to send a request output to the external device, the request output based at least partly on the request from the one access point (figure 6, control signals, explained further in [0045] and [0046]; it is inherent that which of the external devices gets the control signals is based upon the data address or other parameter in the memory access instruction), iii) responsive to the request output, to receive an external reply from the external device ([0040], memory retrieves the data and makes it available through data bus 4020; the newly retrieved

data is the external reply), and iv) to send to the access point a response, based on the external reply, to the request (Figure 1, given the memory controller connects the processor and memories together, it is inherent that the retrieved data would be passed on to the pipeline).

The teaching of Dorst reduces the burden and overhead of processors interfacing with and controlling the memory (Dorst, [0003]), as well as flexibly controls a multitude of memory circuits in a simple-to-use manner (Dorst, [0006]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dorst with the invention of Cataldo in order to reduce the burden and overhead of processors interfacing with and controlling the memory, as well as flexibly control a multitude of memory circuits in a simple-to-use manner. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the teaching of Dorst is applicable to the invention of Cataldo; as cited above in [0031], Dorst discloses that his teaching may be directed toward multiple processors, and Dorst additionally discloses in [0036] that the teaching is applicable to multiprocessing or distributed processing systems. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that Cataldo's specific multiprocessing architecture being a pipeline of processors would not affect Dorst's teaching as each processor of Cataldo would be able to be a processor of Dorst's teaching. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that Dorst's teaching when applied to the pipelined architecture of Cataldo would cause the request to be sent to the interface engine when

the data packet to be serviced reaches the processor that is making the memory request.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dorst with the invention of Cataldo in order to reduce the burden and overhead of processors interfacing with and controlling the memory, as well as flexibly control a multitude of memory circuits in a simple-to-use manner.

Dorst discloses in [0031] that several processors may share a memory controller as persons of ordinary skill in the art would understand; however, both Cataldo and Dorst also do not explicitly disclose that the interface engine includes an arbiter configured to allow forwarding of requests from the plural access points in a fair manner between each of the plural access points.

Although the examiner believes it would have been readily recognized to one of ordinary skill in the art at the time of the invention that there must be some form of arbitration given that multiple processors are sharing a memory controller which cannot handle all their requests at one time, Harriman nevertheless discloses of an arbiter configured to allow forwarding of requests from the plural access points in a fair manner between each of the plural access points (col. 4, lines 19-40, arbiter 224, round robin scheme).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Harriman with the invention of Cataldo and Dorst in order to provide equal access to different requesters (Harriman, col. 4, lines 29-32). It

would have been readily recognized to one of ordinary skill in the art at the time of the

invention that the teaching of Harriman fits into the environment of Cataldo and Dorst:

like Cataldo and Dorst, the plural access points of Harriman are processing logic and

the requests are memory requests.

Therefore, it would have been obvious to one of ordinary skill in the art at the

time of the invention to combine the teaching of Harriman with the invention of Cataldo

and Dorst in order to provide equal access to different requesters.

7.   Consider claims 2 and 9, Dorst discloses the request comprises a first request

code according to a first coding scheme (it is inherent that a data instruction such as a

load or store has parameter bits to indicate the address or memory module that needs

to be accessed; these bits correlate to the request code and its format to be recognized

is the coding scheme), the interface engine being adapted to execute a program ([0070]

and [0071] discloses using finite state machines, counters, and programmable registers

in order to implement the control circuitry for the memory controller and establish

relative timing relationships among control signals and address signals, which is in

effect executing a program), the execution being dependent upon the first request code

([0071] discloses that programmable registers provide a mechanism for timing among

control signals, and [0039] specifically discloses that the programmable registers store

read timing-parameters and write timing-parameters. Because [0042] discloses that

each memory may correspond to a respective register set, it is inherent that, once a

memory instruction occurs and its relevant parameters are sent to the memory

controller, the bits regarding the specific address or memory to be accessed are used in

some form to select the register with the timing-parameters relevant to that specific

address or memory. These timing-parameters are used by the finite state machine to

execute the program, as certain control signals are asserted at certain times to certain

memory because of the parameters/code of the memory instruction), and to obtain, as a

result of the execution of the program, at least one device control code, according to a

second coding scheme (succinctly stated in the third sentence of the abstract, the

interface circuitry communicates with the memory by providing a plurality of control

signals; as explained above, certain control signals, correlating to device control code,

are asserted at certain times as a result of executing the finite state machine that

correlates to the memory bring accessed. This second coding scheme is the code that

is outputted to the memory which controls the memory, as opposed to the first coding

scheme which is the code that makes up the instruction), in addition to which the

interface engine is adapted to send the device control code to the external device, or

the request output is based at least partly on the device control code ([0045], the

memory controllers control a wide variety of memories with the control signals described

in [0046]).

8.   Consider claims 3 and 10, Dorst discloses the device control code is an

operational code of the external device ([0046], the control signals serve to operate the

external device).

9.   Consider claims 4 and 11, Dorst discloses the program is stored in a microcode

memory included in the interface engine ([0070], finite state machine; it is inherent that

the finite state machine contains a register to store state variables, and the state

variables determine which control signals (in essence, microinstructions that control an

overall memory machine instruction) are outputted given that state).

10.   Consider claims 5 and 12, Dorst discloses the pipeline comprises a plurality of

access points ([0031], several processors may share a memory controller; each

processor is an access point), and the interface engine is adapted to receive a request

from at least one of the access points ([0031], several processors may share a memory

controller; each processor is an access point), the interface engine comprising a reply

control unit (figure 3, memory controller 1005 still) adapted to receive at least one

receiver ID signal related to the request, and to determine, based on the receiver ID

signal, the access point which is to receive the response (it is inherent that given a

plurality of processors sharing the same memory controller, a processor which requests

data from memory for itself would be the one to actually receive data. Because of this,

it is further inherent that there must be an ID signal, either explicitly given by the

processor identifying itself or implicit in that merely sending a memory instruction to the

memory controller is an implicit signal as the memory controller is capable of identifying

which processor sent that memory instruction, perhaps due to the pins at which it was

received on).

11.   Consider claims 6 and 13, Dorst discloses the reply control unit is adapted to

receive an input control signal ([0010], memory controller couples to the processor and

to the memories, and provides communication between the processor and memories; it

is inherent that this communication includes a request for data in a memory by the

processor; this request for data correlates to the input control signal, as the request for

data is inputted into the memory controller in order to control the memory into outputting

the data), based on which timing information for receiving the external reply from the

external device can be determined ([0071] discloses that programmable registers

provide a mechanism for timing among control signals, and [0039] specifically discloses

that the programmable registers store read timing-parameters and write timing-

parameters. Because [0042] discloses that each memory may correspond to a

respective register set, it is inherent that, once a memory instruction occurs and its

relevant parameters are sent to the memory controller, the bits regarding the specific

address or memory to be accessed are used in some form to select the register with the

timing-parameters relevant to that specific address or memory. These timing-

parameters are used by the finite state machine to execute the program, as certain

control signals are asserted at certain times to certain memory because of the

parameters/code of the memory instruction. One of these control signals, as seen in

Table 1 near [0053], is read-enable pulse-width, which is used to program the number

of cycles the read-enable signal remains asserted during read operations).

12.   Consider claims 7 and 14, Dorst discloses the pipeline comprises a plurality of access points ([0031], several processors may share a memory controller; each processor is an access point), whereby the number of access points adapted to send a request to the interface engine can be adjusted ([0031], which first discloses that a system may have more than one processor and more than one memory controller. It then discloses that several processors *may* share a memory controller. If several processors *may* share a memory controller, then it is possible that several processors *do not* share a memory controller; and if a given processor is not able to access all memory controllers, then it is apparent that they cannot send a request to that memory controller either. Furthermore, because it is disclosed that several processors *may* share a memory controller, or vice-versa, it is inherent that which is actually the case is adjustable).

13.   Consider claims 15, 16, and 20, Harriman discloses the fair manner of the arbiter is a round-robin manner between the plurality of access points (col. 4, lines 19-40, arbiter 224, round robin scheme).

14.   Consider claim 19, although Dorst discloses in [0031] that several processors may share a memory controller as persons of ordinary skill in the art would understand; however, both Cataldo and Dorst also do not explicitly disclose that the interface engine includes an arbiter configured to allow forwarding of requests from the plurality of access points in a fair manner between each of the plurality of access points.

Although the examiner believes it would have been readily recognized to one of ordinary skill in the art at the time of the invention that there must be some form of arbitration given that multiple processors are sharing a memory controller which cannot handle all their requests at one time, Harriman nevertheless discloses of an arbiter configured to allow forwarding of requests from the plurality of access points in a fair manner between each of the plurality of access points (col. 4, lines 19-40, arbiter 224, round robin scheme).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Harriman with the invention of Cataldo and Dorst in order to provide equal access to different requesters (Harriman, col. 4, lines 29-32). It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the teaching of Harriman fits into the environment of Cataldo and Dorst: like Cataldo and Dorst, the plural access points of Harriman are processing logic and the requests are memory requests.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Harriman with the invention of Cataldo and Dorst in order to provide equal access to different requesters.

15.   Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cataldo in view of Dorst.

16.   Consider claim 17, Cataldo discloses a programmable pipeline with a plurality of

access points located in a spaced-apart relation between a first end of the pipeline and

a second end of the pipeline (Page 1, second to fifth paragraphs, programmable

pipeline architecture...PISC can be programmed to do tasks...each PISC stage acts as

a mini-processor; these access points are each processor or its corresponding ports),

the pipeline configured to directionally transfer data packets from the first end to the

second end (Page 1, second to fifth paragraphs, programmable pipeline

architecture...PISC can be programmed to do tasks...each PISC stage acts as a mini-

processor);

         However, Cataldo does not disclose of an interface engine, having a reply control

unit, connected to an external device located externally of the processor, and connected

to each of the plurality of access points, the interface engine configured to i) receive a

request from the pipeline upon arrival of a data packet at any one of the plurality of

access points, ii) send a request output to the external device, the request output based

at least partly on the request, iii) responsive to the request output, receive an external

reply from the external device, and iv) send a response, based on the external reply, to

the pipeline, and the reply control unit configured to i) receive a receiver ID signal

related to the request, and ii) determine, based on the receiver ID signal, the any one of

the plurality of access points of the pipeline to receive the response .

         On the other hand, Dorst does disclose at least one interface engine (figure 3,

memory controller 1005, which includes interface circuitry 4010 as shown in figure 4),

having a replay control unit (figure 3, memory controller 1005 still), connected to at least

one external device located externally of the processor (figure 3 shows multiple

memories 1015 external to the processor and connected to the data processing block

3005, which includes the memory controller), and connected to each of the plurality of

access points, ([0031], several processors may share a memory controller; each

processor is an access point or contains an access point), the interface engine

configured to i) receive a request from one of the plurality of access points ([0010],

memory controller couples to the processors and to the memories, and provides

communication between the processor and memories; it is inherent that this

communication includes a request for data in a memory by the processor), ii) send a

request output to the external device, the request output based at least partly on the

request (figure 6, control signals, explained further in [0045] and [0046]; it is inherent

that which of the external devices gets the control signals is based upon the data

address or other parameter in the memory access instruction), iii) responsive to the

request output, receive an external reply from the external device ([0040], memory

retrieves the data and makes it available through data bus 4020; the newly retrieved

data is the external reply), and iv) send a response, based on the external reply, to the

access point (Figure 1, given the memory controller connects the processor and

memories together, it is inherent that the retrieved data would be passed on to the

access point), and the reply control unit configured to i) receive a receiver ID signal

related to the request, and ii) determine, based on the receiver ID signal, the any one of

the plurality of access points of the pipeline to receive the response (it is inherent that

given a plurality of processors sharing the same memory controller, a processor which

requests data from memory for itself would be the one to actually receive data.
Because of this, it is further inherent that there must be an ID signal, either explicitly
given by the processor identifying itself or implicit in that merely sending a memory
instruction to the memory controller is an implicit signal as the memory controller is
capable of identifying which processor sent that memory instruction, perhaps due to the
pins at which it was received on).

The teaching of Dorst reduces the burden and overhead of processors
interfacing with and controlling the memory (Dorst, [0003]), as well as flexibly controls a
multitude of memory circuits in a simple-to-use manner (Dorst, [0006]).

It would have been obvious to one of ordinary skill in the art at the time of the
invention to combine the teaching of Dorst with the invention of Cataldo in order to
reduce the burden and overhead of processors interfacing with and controlling the
memory, as well as flexibly control a multitude of memory circuits in a simple-to-use
manner. It would have been readily recognized to one of ordinary skill in the art at the
time of the invention that the teaching of Dorst is applicable to the invention of Cataldo;
as cited above in [0031], Dorst discloses that his teaching may be directed toward
multiple processors, and Dorst additionally discloses in [0036] that the teaching is
applicable to multiprocessing or distributed processing systems. It would have been
readily recognized to one of ordinary skill in the art at the time of the invention that
Cataldo's specific multiprocessing architecture being a pipeline of processors would not
affect Dorst's teaching as each processor of Cataldo would be able to be a processor of
Dorst's teaching. It would have been readily recognized to one of ordinary skill in the art

at the time of the invention that Dorst's teaching when applied to the pipelined

architecture of Cataldo would cause the request to be sent to the interface engine when

the data packet to be serviced reaches the processor that is making the memory

request.

Therefore, it would have been obvious to one of ordinary skill in the art at the

time of the invention to combine the teaching of Dorst with the invention of Cataldo in

order to reduce the burden and overhead of processors interfacing with and controlling

the memory, as well as flexibly control a multitude of memory circuits in a simple-to-use

manner.

17.   Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cataldo

and Dorst as applied to claim 17 above, and further in view of Berggreen (US

20030016685 A1).

18.   Consider claim 18, neither Cataldo nor Dorst disclose a quantity of the plurality of

access points from which the request is received by the interface engine from the

pipeline is adjustable based on one of i) a data flow rate through the pipeline, and ii) a

flow capacity of the external device.

On the other hand, Berggreen does disclose of the concept of processing

packets in different manners depending on one of i) a data flow rate through the pipeline

and ii) a flow capacity of the external device ([0038]-[0039], for example, a first

operating mode is used when the data flow rate is low and a second operating mode is

used when the data flow rate is high).

Berggreen's teaching increases flexibility by catering to a wide range of

configurations and environments in part on current and historical operating conditions

while reducing congestion and maintaining QoS for all message types (Berggreen,

[0006]-[0007]).

It would have been obvious to one of ordinary skill in the art at the time of the

invention to combine the teaching of Berggreen with invention of Cataldo and Dorst in

order to increase flexibility by catering to a wide range of configurations and

environments in part on current and historical operating conditions while reducing

congestion and maintaining QoS for all message types. It would have been readily

recognized to one of ordinary skill in the art at the time of the invention that Bergreen's

teaching of processing packets in differing ways dependent on data flow rate, when

applied to the invention of Cataldo and Dorst, encompasses the possibility that a

memory access needed for one mode is not needed for another. Additionally, the

second mode of Berggreen involves storing received messages in a second queue

which does not happen in the first mode, which is one example of how rate would

determine whether a memory access occurs from a specific access point of the

programmable pipeline or not. It would have been readily recognized to one of ordinary

skill in the art at the time of the invention that the environment of Berggreen regarding

QoS is analogous to the invention of Cataldo which can be programmed to do traffic

shaping.

Therefore, it would have been obvious to one of ordinary skill in the art at the

time of the invention to combine the teaching of Berggreen with invention of Cataldo

and Dorst in order to increase flexibility by catering to a wide range of configurations
and environments in part on current and historical operating conditions while reducing
congestion and maintaining QoS for all message types.

Response to Arguments

19.   Applicant argues on page 12 that Dorst does not teach or suggest a
programmable pipeline. However, Cataldo teaches this limitation. Applicant argues
that Cataldo does not teach plural access points; however, Dorst teaches of processors
connected to memory, wherein the processors are then access points as access to
memory originates at the processors. It is also noted Cataldo does teach plural access
points in that the processors access the data that is traversing the pipeline. Therefore,
Cataldo as modified by Dorst teaches that the plural access points are located in a
spaced apart relation between the first end of the pipeline and the second end of the
pipeline.

20.   Applicant argues on page 13 that an access point as recited by the claims would
cause the pipeline of Cataldo to wait, halting the pipeline while the interface engine
processes the request, thereby forfeiting the inventive feature offered by Cataldo of
processing network packets at wire speed. However, this is not necessarily true, such
as when the access to memory takes up less time than the pipeline and wire speed.
Applicant argues there is no suggestion in Cataldo that its sequential pipelined
processor would benefit or interface with Dorst's memory controller; however, Dorst

gives motivation for using his memory controller, and there is no explicit teaching in

either reference, nor any implicit reason known to one of ordinary skill in the art at the

time of the invention, as to why the memory controller of Dorst and the pipelined

processor of Cataldo could not be combined.

21.   Applicant argues that neither Cataldo nor Dorst disclose the newly amended

limitations of an arbiter; in response, examiner has brought in another art as explained

in the rejection above.

Conclusion

22.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Keith Vicary whose telephone number is (571)270-1314.

The examiner can normally be reached on Monday - Thursday, 6:15 a.m. - 5:45 p.m.,

EST.

        If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

K. Vicary